CSA 250: Deep Learning Project IV Report

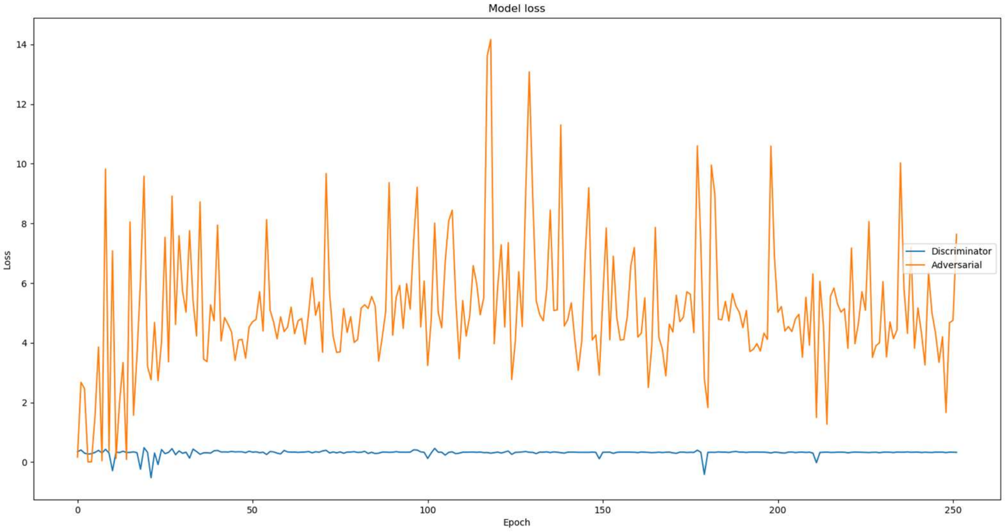
Modukuri Sumanth Kumar

M.Tech AI sumanthkumar@iisc.ac.in S/RNo.: 17025

Tools Used:

- Python 3
- Tensorflow 2.x
- Keras
- Tensorflow-gan

Epochs vs Loss:



Questions:

- Describe what SAGAN is and how it differs from a DCGAN.

Self-Attention GANs have an architecture that allows the generator to model long-range dependency. The key idea is to enable the generator to produce samples with global detailing information. It has an added Attention layer after a convolutional layer in the generator and the discriminator. The output of the previous convolution layer outputs convolution feature maps of the dimension.

DCGAN is useful in finding local features, but SAGAN helps by giving a global view as well. It help generalize the generator.

- explain what spectral normalization is.

  GAN is vulnerable to mode collapse and training instability. Spectral normalization solves this issue by normalizing the weight for each layer with the spectral norm σ(W) such that the Lipschitz constant for each layer as well as the whole network equals one.

- what attention is and why it is beneficial to many machine learning models.

  Attention is simply a vector, often the outputs of dense layer using softmax function. It allows machine translator to look over all the information the original sentence holds, then generate the proper word according to current word it works on and the context. It can even allow translator to zoom in or out (focus on local or global features).

Key Points:

- I was not able to train the GAN model properly, so the generator generates the images in which few have similar patterns as it just learned the manifold of what it had to learn.

- The generator loss is more erratic than discriminator as discriminator is supposed to be the truth and generator is learning to close the gap.
- Using different distributions for seed gives different convergence rates.
- DCGAN works well as it uses CNNs to learn and since we deal with images as dataset, and CNNs are great at finding features on images.
- Tried implementing FID, but was not able to merge it with Keras training method.
- I was not able to add the self-attention layer to the Keras model.
- Did try hyperparameter tuning using random search as the computing resources were not sufficient. Tried using the Clserv but it was not working properly, so had to implement all on Google Colab.